GENERÁTOR ČTYŘÚHELNÍKOVÉ SÍTĚ

QUADRILATERAL MESH GENERATOR

Petr Frantík[1]

Abstrakt

Článek pojednává o generování sítě čtyřúhelníků pro potřeby metody konečných prvků. Síť je generována na libovolné množině bodů, pomocí Deloneho triangulace. Tvar sítě je optimalizován pomocí dynamické simulace.

Klíčová slova

Čtyřúhelníková síť, Deloneho triangulace, tvarová optimalizace, dynamická simulace.

Abstract

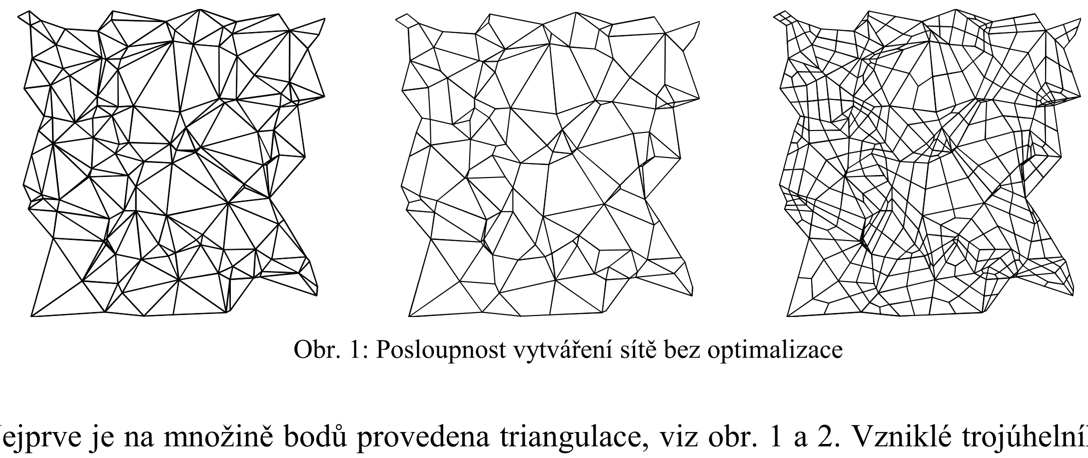
The paper is focused on making meshes of quadrilaterals for application of finite element method. The mesh is generated on arbitrary set of points with help of the triangulation Delaunay. The shape of the mesh is optimized by a dynamical simulation.

Keywords

Quadrilateral mesh, triangulation Delaunay, shape optimization, dynamical simulation.

1 Úvod

Čtyřúhelníkové sítě se intenzivně užívají pro výpočty deformací a napětí v rovinných úlohách mechaniky deformovatelných těles pomocí metody konečných prvků. Každý čtyřúhelník sítě by měl být tvarově co nejbližší čtverci z důvodu výstižnosti aproximovaných veličin a numerické stabilitě. Zde prezentovaný postup vytváření je založen na pěti jednotlivých technikách použitelných samostatně.

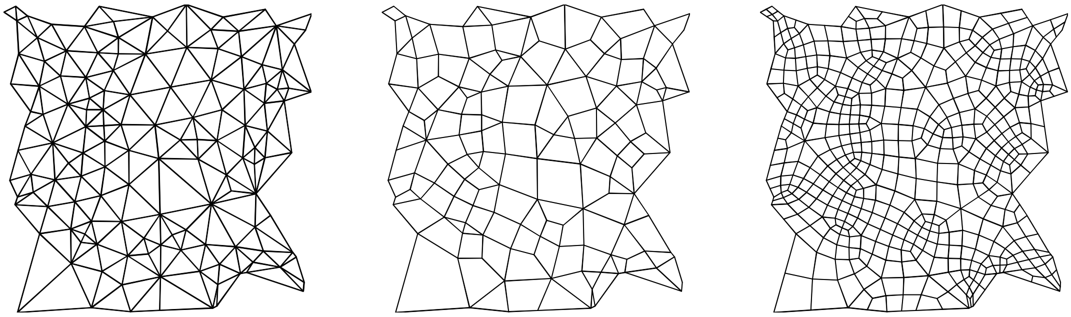


Obr. 1: Posloupnost vytváření sítě bez optimalizace

Nejprve je na množině bodů provedena triangulace, viz obr. 1 a 2. Vzniklé trojúhelníky jsou poté spojovány do čtyřúhelníků, dokud nezůstanou trojúhelníky, které již nemají

[1] Ing. Petr Frantík, Ph.D., Vysoké učení technické v Brně, Fakulta stavební, Ústav stavební mechaniky, Veveří 331/95, 602 00 Brno, e-mail: kitnarf@centrum.cz

vhodné partnery ke spojení. Vzniklá smíšená síť se rozdělí tak, že vzniknou čtyřúhelníky s přibližně dvojnásobnou hustotou. Dále se odstraní specifické kosočtverečné útvary. V každé fázi se přitom provádí optimalizace tvaru sítě, která ovšem mění polohy uzlů.



Obr. 2: Posloupnost vytváření sítě s optimalizací

2 Triangulace

K vytvoření trojúhleníkové sítě i náhodně generovaných bodů je užito Deloneho triangulace. Tato triangulace zaručuje, že v kružnici opsané trojici vrcholů každého trojúhelníka neleží žádný jiný bod, viz např. [1]. Současně maximalizuje minimální úhel mezi stranami trojúhelníkové sítě, viz obr. 1 vlevo.

Po provedení triangulace lze síť podrobit optimalizaci (obr. 2 vlevo), která bude souhrnně popsána v poslední kapitole.

3 Spojování trojúhelníků

V této fázi se prochází všechny trojúhelníky. Pro každý trojúhelník je nalezena trojice sousedních trojúhelníků. Pro tři čtyřúhelníky vzniklé spojením původního trojúhelníka s jedním z partnerů je spočítáno bezrozměrné číslo, tzv. index pravidelnosti (regularity) r, pro které platí:

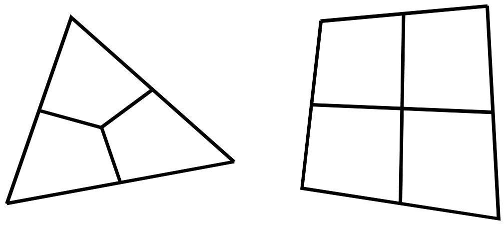
$$r = \frac{A}{C^2}, \tag{1}$$

kde A je plocha čtyřúhelníka a C je jeho obvod. Spojení partnerů se provede pro čtyřúhelník s nejvyšším indexem pravidelnosti, pokud tento index překročí určitou prahovou hodnotu (v Java balíku [2] je prahová hodnota nastavena na 0.04; čtverec má index $r = 1/16$, tj. 0.0625). Výsledkem je smíšená síť obsahující trojúhelníky a čtyřúhelníky, viz obr. 1 uprostřed.

Po provedení spojování lze síť opět podrobit optimalizaci (obr. 2 uprostřed).

4 Rozdělení

Síť s čtyřúhelníky získáme z předchozího výsledku tím, že vytvoříme nové uzly ve středu každého n-úhelníka a ve středech všech hran sítě [3]. Každý trojúhelník respektive čtyřúhelník se rozpadne na tři respektive čtyři čtyřúhelníky, viz obr. 3.

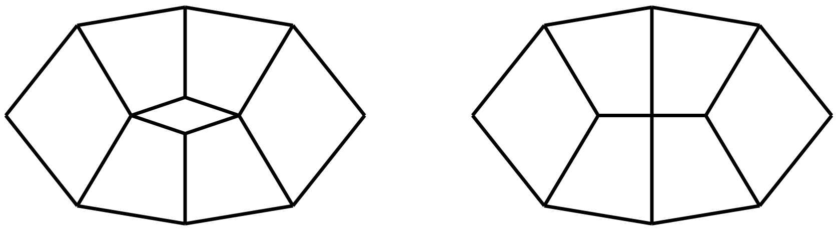


Obr. 3: Rozdělení n-úhelníků

Po provedení rozdělení lze síť znovu podrobit optimalizaci (obr. 2 vpravo).

5 Odstranění kosočtverců

Výše popsaný postup generování sítě typicky vytváří nevhodné kosočtverečné úvary znázorněné na obrázku 4. Tento útvar lze snadno detekovat (v balíku [2] je užito speciálního grafu) a odstranit sloučením bližších protilehlých uzlů kosočtverce.



Obr. 3: Kosočtverečný útvar a způsob jeho odstranění

6 Optimalizace tvaru

Tvar n-úhelníkové sítě je optimalizován technikou navrženou pro trojúhelníkové sítě, viz [4]. Obecná n-úhelníková síť je transformována na trojúhelníkovou rozdělením každého n-úhelníka na trojúhelníky přidáním uzlu do středu.

Tato trojúhelníková síť je dále chápána jako množina hmotných bodů (o jednotkové hmotnosti) propojená pružinami jednotkové tuhosti. Délky pružin jsou nastaveny podle aritmetických průměrů délek obvodových hran n-úhelníků. Délka pružiny na obvodu původního n-úhelníku činí $(L_1+L_2)/20$, kde L_1 respektive L_2 je průměrná délka pružin levého resp. pravého n-úhelníka. Délky pružin uvnitř n-úhelníku mají délku $L/20$, kde L je průměrná délka hran n-úhelníka. Dělení desíti zajišťuje předpětí sítě.

Takto vytvořená pružinová síť je poté chápána jako dynamický systém:

$$\begin{aligned} \frac{dv_x}{dt} &= R_x - v_x, \quad \frac{dx}{dt} = v_x, \\ \frac{dv_y}{dt} &= R_y - v_y, \quad \frac{dy}{dt} = v_y, \end{aligned} \qquad (2)$$

kde x a y jsou souřadnice hmotného bodu, v_x a v_y jsou složky jeho rychlosti, R_x a R_y jsou složky výslednice sil od pružin a t je čas. Systém je řešen pomocí symplectické Eulerovy metody s krokem $h = 0.1$ s:

$$
\begin{aligned}
v_x(t+h) &= v_x(t) + h(R_x(t) - v_x(t)), \\
v_y(t+h) &= v_y(t) + h(R_y(t) - v_y(t)), \\
x(t+h) &= x(t) + hv_x(t+h), \\
y(t+h) &= y(t) + hv_y(t+h),
\end{aligned}
\tag{2}
$$

Pro ustálení kmitání systému postačuje provedení 100 kroků.

7 Závěr

Článek se věnoval metodě generování sítě čtyřúhelníků pro potřeby metody konečných prvků vhodné i pro náhodně generované uzly. Podobný princip půjde pravděpodobně aplikovat i pro trojrozměrné sítě.

Prezentované výsledky byly získány za finanční podpory projektu specifického vysokoškolského výzkumu registrovaného na VUT v Brně pod č. FAST-S-12-34.

Literatura

[1] SHEWCHUK, J. R., Triangle: Engineering a 2d quality mesh generator and delaunay triangulator, *Applied computational geometry: Towards geometric engineering*, M. C. Lin and D. Manocha, eds., vol. 1148, Springer-Verlag, 1996, pp. 203–222.

[2] FRANTÍK, P., Java balík `cz.kitnarf.graph`, uveřejněno na Internetu na adrese `http://kitnarf.cz/java`, 2007

[3] BOROUCHAKI, H. a FREY, P. J., Adaptive triangular--quadrilateral mesh generation, *International journal for numerical methods in engineering*, vol. 41, 915-934, 1998

[4] PERSSON, P.-O. a STRANG, G., A simple mesh generator in Matlab, SIAM Review, Volume 46 (2), pp. 329-345, June 2004